

UNIVERSITY OF MICHIGAN

# EECS 270: Intro to Logic Design

## Final Exam

Prof. Karem A Sakallah

Wednesday December 18, 2019  
4:00-6:00 p.m.

A - R: 220 CHRYSLER || S - Z: 906G COOLEY

Name:	_____
UMID:	_____
<b>Honor Pledge:</b> "I have neither given nor received aid on this exam, nor have I concealed any violation of the Honor Code."	
Signature: _____	

**Instructions:**

- The exam is closed book except for three 8.5"x11" sheets of notes. No electronics of any kind may be used.
- Print your name and student ID number and sign the honor pledge.
- The exam consists of 6 problems with the point distribution indicated here. Please keep this in mind as you work through the exam. Use your time wisely.
- There are 11 sheets in this exam. Make sure that you have all 11 sheets and notify an instructor if you do not.

1.	/15
2.	/15
3.	/20
4.	/25
5.	/15
6.	/10
Total:	/100

## 1 [You ought to know this stuff :-)-15 points]

a. [1.5 points]

When adding two  $n$ -bit signed numbers, overflow can never occur if the sign bits of the addends are the same.

TRUE  FALSE

b. [1.5 points]

All minterms of a function are prime implicants.

TRUE  FALSE

c. [1.5 points]

A minterm that is a prime implicant must be essential.

TRUE  FALSE

d. [1.5 points]

If  $ab'$  is an implicant of a function  $f$ , then  $ab'd$  is too.

TRUE  FALSE

e. [1.5 points]

A positive level-sensitive D latch stores the value present on the D input on the falling edge of the clock.

TRUE  FALSE

f. [1.5 points]

The output of an SR latch will oscillate while  $S = 1$  and  $R = 1$ .

TRUE  FALSE

g. [1.5 points]

The number of transitions from any state of a sequential circuit that has three single-bit inputs is at most three.

TRUE  FALSE

h. [1.5 points]

A sequential circuit ~~can not~~ have both Moore-type and Mealy-type outputs.

TRUE  FALSE

i. [1.5 points]

In a ~~safe~~ sequential circuit design, all transitions out of an "unused" state must go to valid states.

TRUE  FALSE

j. [1.5 points]

An  $n$ -bit register can be used as the memory element in any sequential circuit with up to  $2^n$  states.

TRUE  FALSE

## 2 [And this stuff too! - 15 points]

$$64 = 2^6 \quad 16 \quad \boxed{?} \quad \boxed{?} \quad \boxed{?} \quad \boxed{?} \quad \boxed{?} \quad \boxed{?}$$

- a. [1.5 points] We wish to make a 64-way 1-bit multiplexor from several 4-way 1-bit multiplexors and no other components. What is the minimum number of 4-way multiplexors we need?

$$\times \checkmark \geq$$

Answer:  $16 + 4 + 1 = 21$    $m_1 \quad 001 \quad x'y'z$

- b. [2 points] Let  $m_i(x_1, \dots, x_n)$  and  $M_j(x_1, \dots, x_n)$  be, respectively, the  $i^{th}$  minterm and  $j^{th}$  maxterm for  $i, j \in [0, 2^n - 1]$ . Assuming that  $i \neq j$ , to what does each of the following expressions evaluate?

$$m_0 + M_i = 1+0 = 1 \quad \text{since } M_i = m_i$$

$$m_i + M_j = 1+1 = 2 \quad \text{X } M_j \text{ minterm is 1 only in row } i \quad M_j \text{ is 1 only if not in row } j$$

$$m_i \cdot M_i = 1 \cdot 0 = 0$$

$$m_i \cdot M_j = 1 \cdot 1 = 1 \quad \text{X } M_j$$

- c. [1.5 points] The decimal value of the 7-bit string  $\underline{1011101}_{\text{base } 2}$   $\frac{1}{2^6} + \frac{1}{2^5} + \frac{1}{2^4} + \frac{1}{2^3} + \frac{1}{2^2} + \frac{1}{2^1} = 29$   $-64 + 16 + 8 + 4 + 1$

As a signed magnitude number is:  $\frac{-29}{11}$

As a two's complement number is:  $\frac{-35}{11}$

As a ones' complement number is:  $\frac{-34}{11}$

- d. [2.5 points]  $f(a, b, c, d) = \sum_{a,b,c,d}(1, 3, 7, 15)$ .

$$1b-4=2$$

The number of minterms  $f$  has in its canonical SOP representation is:

The number of maxterms  $f$  has in its canonical POS representation is:

The prime implicants of  $f$  are:  $a'b'd, bcd, a'cd, ab'd + bcd$

The minimal SOP expression for  $f$  is  $a'b'c'd + ab'c'd + ab'c'd + ab'c'd$ ,

*not unique*

and it is (circle one)

a	b	c	d	cd	ab	bc	ac	abc
0	0	0	1	00	00	01	11	10
0	0	1	1	00	00	01	11	10
0	1	1	1	01	01	11	11	11
1	1	1	1	11	11	11	11	11

NON-UNIQUE  UNIQUE

- e. [1.5 points]

A sequential circuit has four inputs  $x_3, x_2, x_1$ , and  $x_0$ , four outputs  $y_3, y_2, y_1$ , and  $y_0$  and just four D flip-flops. Input  $x_i$  and output  $y_i$  are connected, respectively, to the  $D$  input and  $Q$  output of flip-flop  $i$ .

The number of states this circuit has is:  $2^4 = 16$

The number of state transitions this circuit has is:  $2^6 \times 2^4 = 256$  (16x16).

This circuit is a (circle one)

Moore

Mealy

Machine.

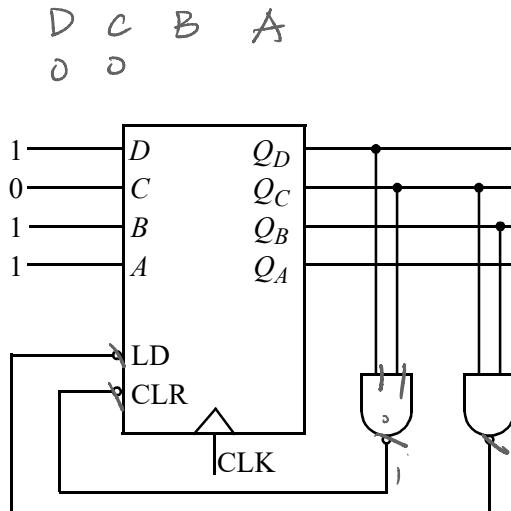
f. [2 points]

00000 + 1010

This circuit uses a binary up-counter with synchronous load and clear inputs. It goes through the same counting sequence over and over again as long as the clock is applied.  $D$  and  $Q_D$  are the most-significant bits of the input and output, respectively. What is the counting modulus?

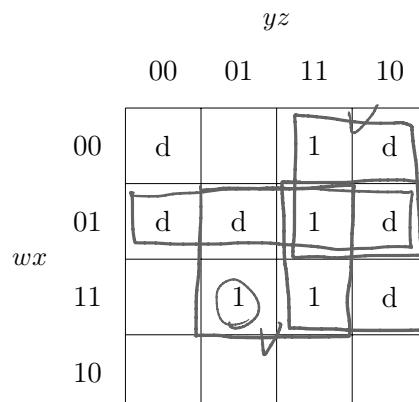
Answer: 9

D	C	B	A
1	0	1	1
1	1	0	0
0	0	0	0
0	0	0	1
0	0	1	0
0	1	1	1
0	1	0	0
0	1	0	1
0	1	1	0
1	0	1	1



g. [2 points]

This K-map is for the incompletely-specified function  $f(w, x, y, z)$ .

The number of  $f$ 's prime implicants is: 4The number of  $f$ 's *essential* prime implicants is: 2

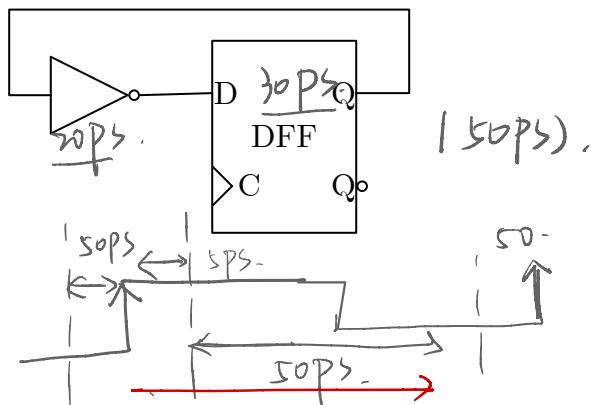
h. [2 points]

50.

The timing parameters for the inverter and D flip-flop in this circuit are:

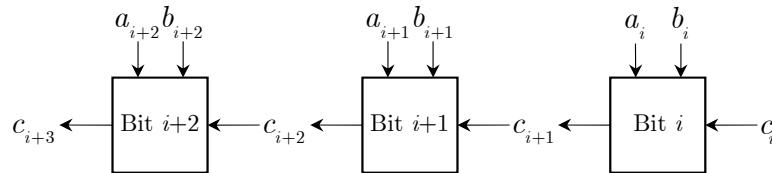
- Inverter delay = 20 ps
- Clock-to-Q delay = 30 ps
- Setup time = 50 ps.
- Hold time = 5 ps

Calculate the minimum clock period for which this circuit will operate without exhibiting metastable behavior.

Answer: 50ps + 50ps + 5ps = 105ps X 50 + 30 + 20 = 100

### 3 [Carry Lookahead Analysis–20 points]

- a. [3 points] Consider the following 3-bit range  $[i+2, i]$  starting at some bit position  $i$ :



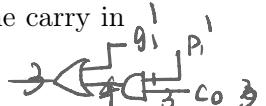
$$P_i = a_i + b_i \quad G_i = a_i \cdot b_i$$

Derive the SOP expressions for the 3-bit group generate and propagate signals  $G_{i+2,i}$  and  $P_{i+2,i}$  in terms of the bit generate and propagate signals  $g_i, g_{i+1}, g_{i+2}, p_i, p_{i+1}$  and  $p_{i+2}$ .

[2 Points]  $G_{i+2,i} = g_{i+2} + P_{i+2} \cdot g_{i+1} + P_{i+2} \cdot P_{i+1} \cdot g_i$

[1 Point]  $P_{i+2,i} = P_{i+2} \cdot P_{i+1} \cdot P_i$

- b. [13 points] Using the following generate/propagate structure of a 9-bit CLA adder, let  $t_P^{i,j}$  be the propagation delay to  $c_j$  given that it is computed from  $c_i$ . Assume that all gates have unit delay and that all inputs are available at  $t = 0$ . Enter these propagation delays in the following table and circle the entries that correspond to the minimum delays. If the carry in column  $j$  cannot be computed from the carry in row  $i$ , cell  $(i, j)$  should be blank.



$c_9$	8	$c_8$	7	$c_7$	6	$c_6$	5	$c_5$	4	$c_4$	3	$c_3$	2	$c_2$	1	$c_1$	0	$c_0$

$$C_1 = g_0 + P_0 \cdot C_0$$

$$C_2 = g_1 + P_1 \cdot C_1$$

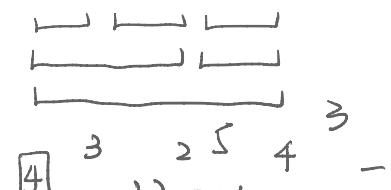
$$C_3 = g_2 + P_2 \cdot C_2$$

$$7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \quad -$$

$$C_3 = G_{2,0} + P_{2,0} \cdot C_0$$

$$G_{2,0} = G_{2,1} + P_{2,1} \cdot g_0$$

$$P_{2,0} = P_0 \cdot P_1 \cdot P_2 \quad 3$$



From  $c_{in}$

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$
$c_0$	(3)		(4)						(6)
$c_1$		(5)							
$c_2$			7						
$c_3$				(6)	(6)				
$c_4$					(8)				
$c_5$						10			
$c_6$							(8)	8	
$c_7$								(10)	
$c_8$									12

$$C_3 = g_2 + P_2 \cdot C_2 \Rightarrow 7$$

$$7 \leftarrow 6 \leftarrow 5 \leftarrow 4 \leftarrow 3 \leftarrow 2 \leftarrow 1 \leftarrow 0$$

$$C_3 = G_{2,0} + P_{2,0} \cdot C_0 \quad 0 \quad 1 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \quad \Rightarrow 5$$

$$G_{2,0} = G_{2,1} + P_{2,1} \cdot g_0 \quad 3 \quad 2 \quad 1 \quad 0 \quad \Rightarrow 5$$

$$G_{2,1} = g_2 + P_2 \cdot g_1 \quad 1 \quad 1 \quad 1 \quad 1 \quad -1$$

$$C_3 = G_{2,0} + P_{2,0} \cdot C_0 \quad 1 \quad 2 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0$$

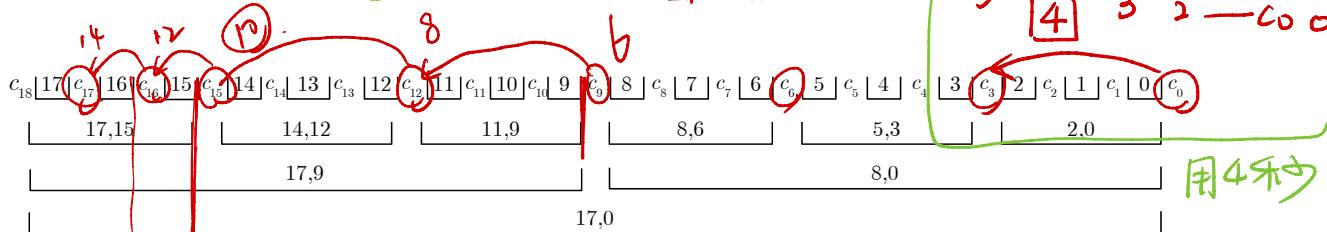
$$G_{3,1} = G_{3,2} + P_{3,2} \cdot g_1$$

$$G_{3,1} = g_3 + P_3 \cdot G_{2,1}$$

为哈

易用

- c. [4 points] Now consider building an 18-bit CLA adder by introducing another level of generate/propagate logic as shown below:



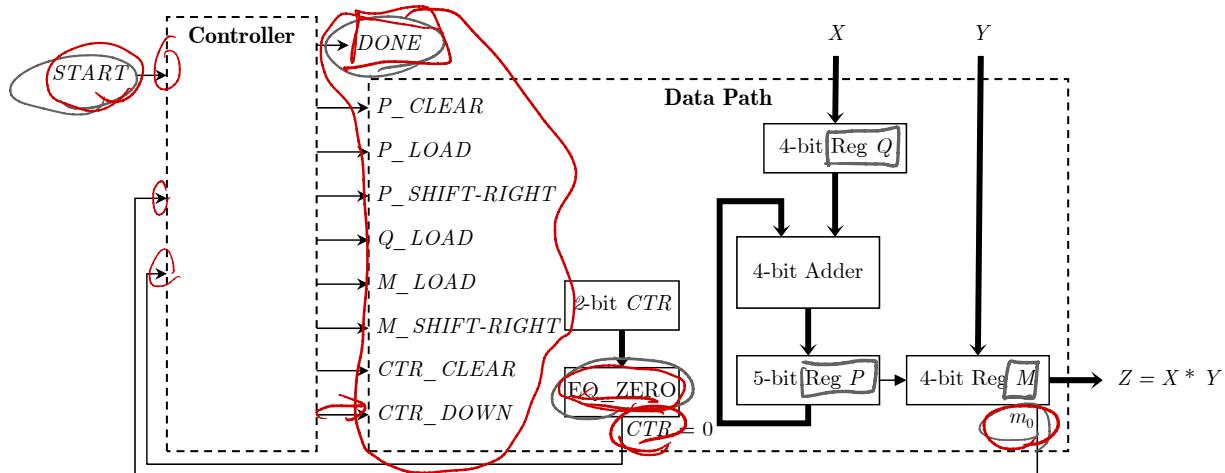
Let  $t_{P,min}^{i,j}$  be the minimum delay from  $c_i$  to  $c_j$ . Compute  $t_{P,min}^{0,17}$  and  $t_{P,min}^{0,18}$ .

[2 Points]  $t_{P,min}^{0,17} = 14$  ✓  $c_9 - c_{17}$  和  $c_0 - c_8$  是一极向.

[2 Points]  $t_{P,min}^{0,18} = 8$  ✓

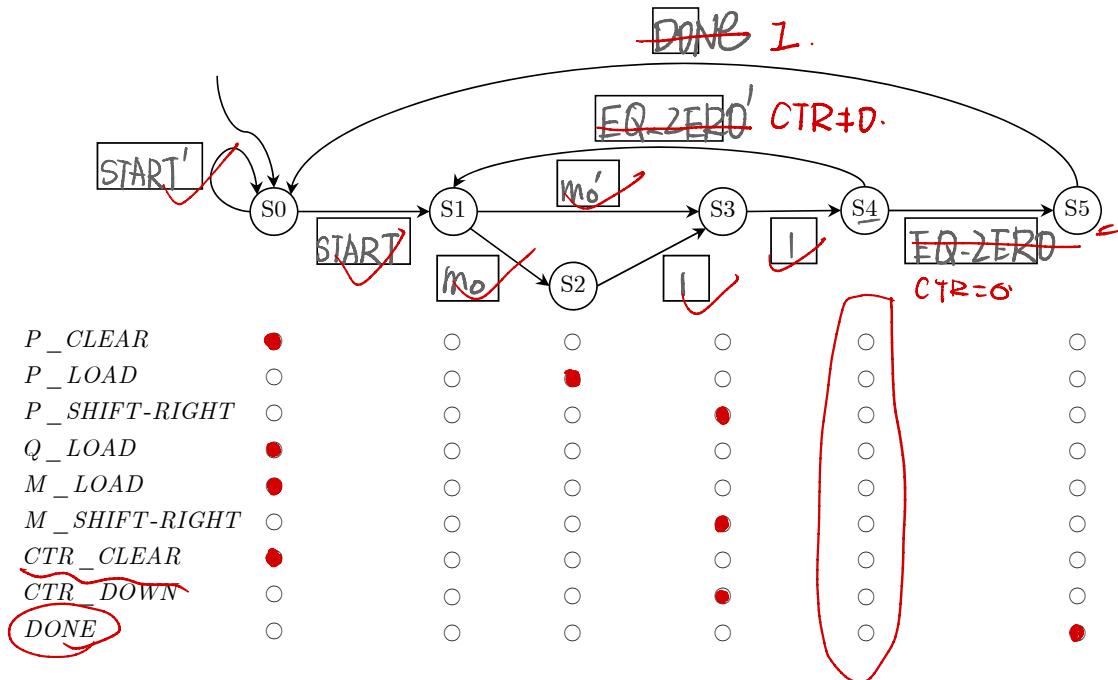
## 4 [Sequential Multiplication–25 points]

Consider the following datapath and controller for an *unsigned* 4-bit sequential multiplier. Its basic structure is similar to design V1.0 that was discussed in lecture but has been augmented to allow for the multiplication of a sequence of unsigned 4-bit numbers. To use it, the *START* “button” must be pressed after the numbers to be multiplied are applied to inputs *X* and *Y*. The result is displayed on the 8-bit output *Z* when the *DONE* indicator turns on. At that point the multiplier returns to its initial state and is ready for another multiplication operation.



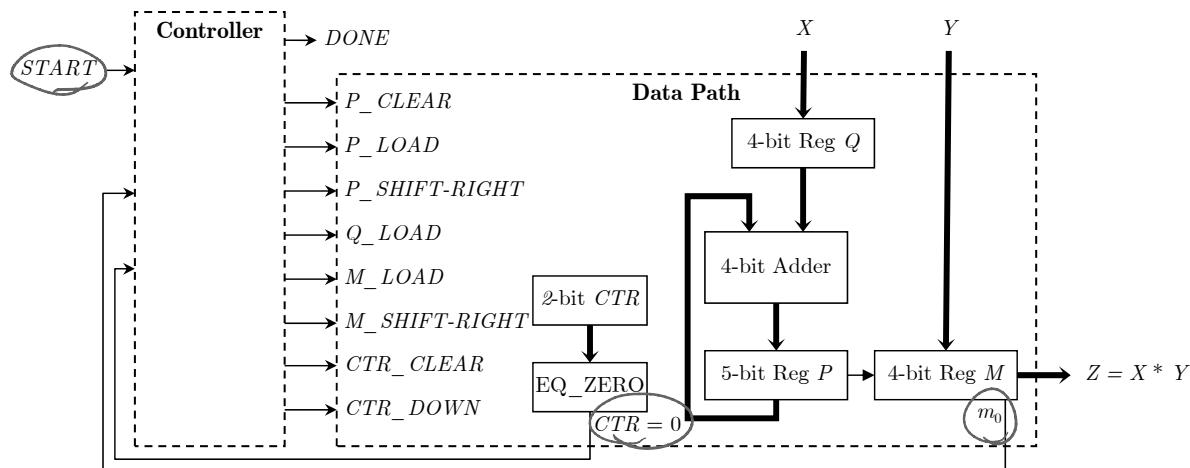
- a. [9 points] Complete the multiplier’s control state diagram, shown below, by:

- Writing the required transition conditions inside the indicated boxes.
- Selecting the outputs that must be set to 1 in each of the controller’s six states by completely filling the appropriate “bubbles”.



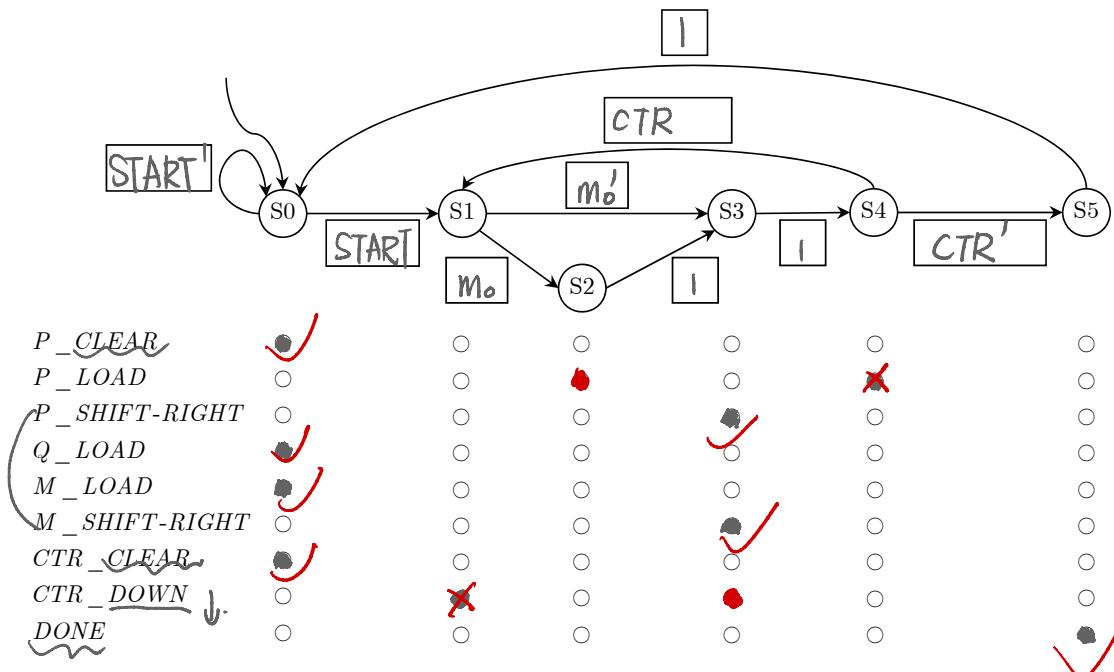
## 4 [Sequential Multiplication–25 points]

Consider the following datapath and controller for an *unsigned* 4-bit sequential multiplier. Its basic structure is similar to design V1.0 that was discussed in lecture but has been augmented to allow for the multiplication of a *sequence* of unsigned 4-bit numbers. To use it, the *START* “button” must be pressed after the numbers to be multiplied are applied to inputs *X* and *Y*. The result is displayed on the 8-bit output *Z* when the *DONE* indicator turns on. At that point the multiplier returns to its initial state and is ready for another multiplication operation.



- a. [9 points] Complete the multiplier’s control state diagram, shown below, by:

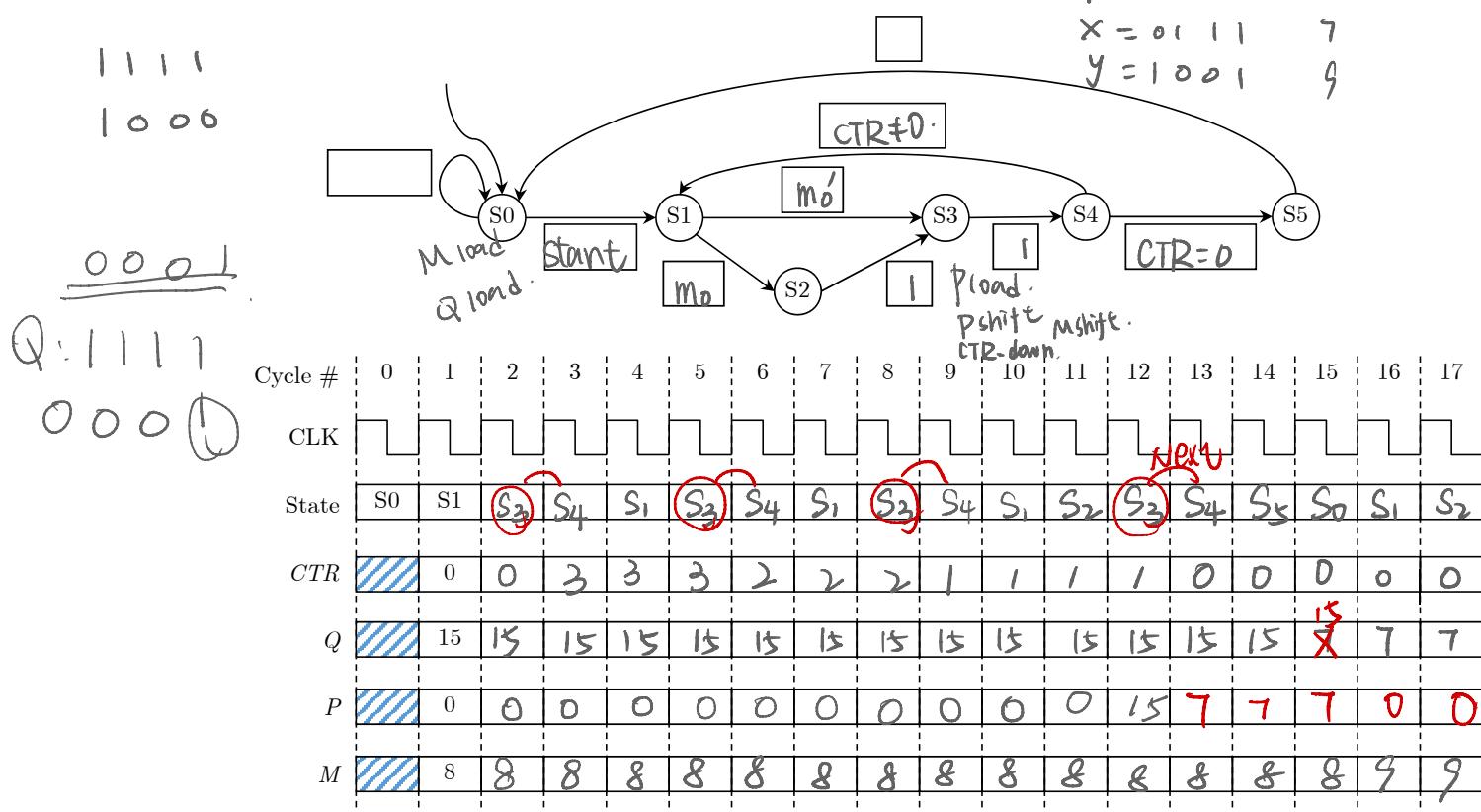
- Writing the required transition conditions inside the indicated boxes.
- Selecting the outputs that must be set to 1 in each of the controller’s six states by completely filling the appropriate “bubbles”.



- b. [16 points] Complete the following timing diagram by indicating the controller state and numeric contents, in decimal, of  $CTR$ ,  $Q$ ,  $P$  and  $M$  in clock cycles 2 to 17 assuming that:

- $START = 1$
- $X = 15$  and  $Y = 8$ , followed by  $X = 7$  and  $Y = 9$

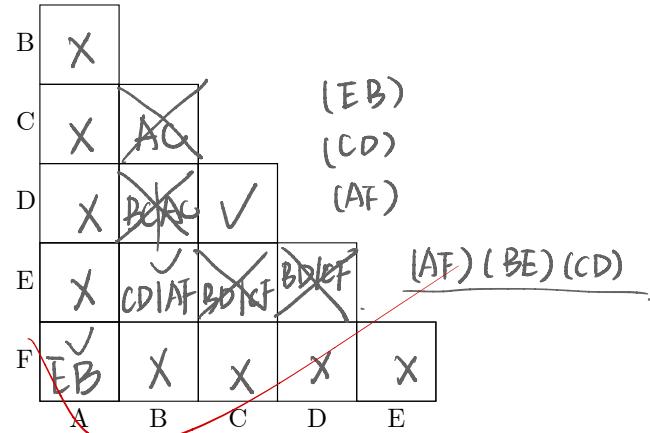
$$\begin{aligned} X &= 1111 & 15 \\ Y &= 1000 & 8 \\ X &= 0111 & 7 \\ Y &= 1001 & 9 \end{aligned}$$



## 5 [State Minimization–15 points]

The state table, shown below, is for a sequential circuit with one input  $x$  and one output  $z$ . Complete the state merger diagram for this circuit and answer the following questions.

Current State	Input		Next State, $z$
	$x = 0$	$x = 1$	
A	A, 1	E, 0	
B	C, 0	A, 1	
C	B, 0	C, 1	
D	B, 0	C, 1	
E	D, 0	F, 1	
F	A, 1	B, 0	
			Next State, $z$



- a. [2 points] States C and E are equivalent.

TRUE  FALSE

- b. [2 points] States A and F are equivalent.

TRUE  FALSE

- c. [2 points] State B is equivalent to:

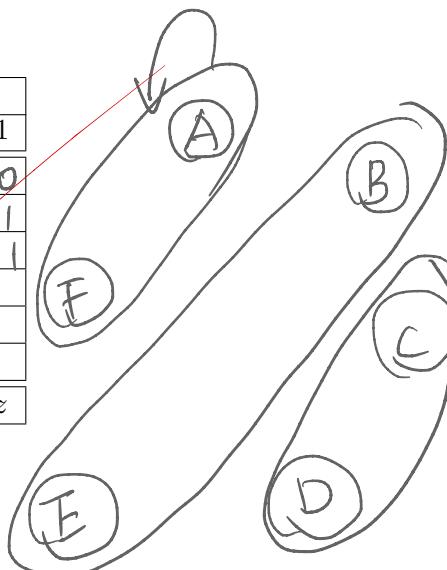
A: State F      B: State C      C: State A      D: State D      E: None of the choices

- d. [2 points] The minimum number of states is:

A: 3      B: 4      C: 5      D: 2      E: 6

- e. [2 points] Using suitable labels for the equivalent states, complete the following minimal state table.

Current State	Input		Next State, $z$
	$x = 0$	$x = 1$	
AF	AF, 1	BE, 0	
BE	CD, 0	AT, 1	
CD	BE, 0	CD, 1	
			Next State, $z$



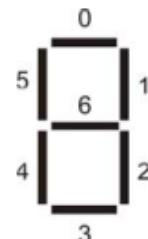
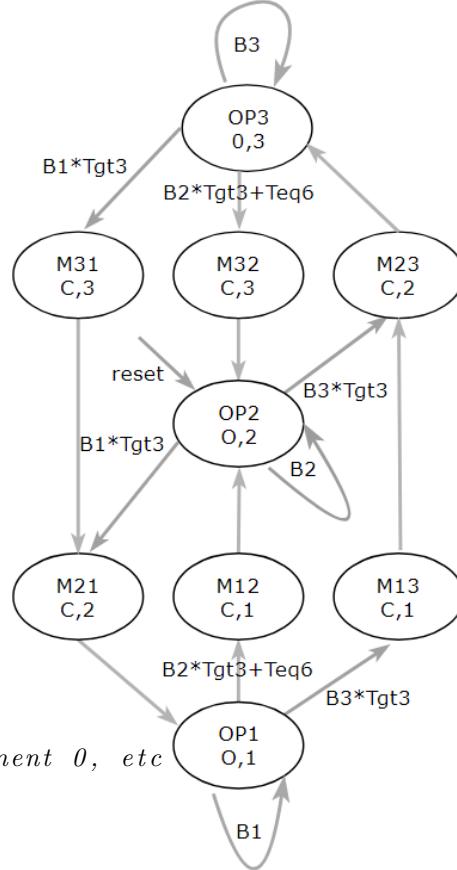
## 6 [Lab Experience—10 points]

The adjacent state diagram represents the operation of an elevator servicing 3 floors. The elevator normally rests on floor 2 with the doors open. The elevator will move to a requested floor if the respective button is pushed and the doors have been open for at least 4 seconds. If the elevator is at any floor for 6 seconds, it will return to floor 2 and remain there until there is another request.

Complete the following Verilog that implements the elevator FSM. You can assume that only one button is pushed at a time and the clock occurs once a second. Assume the state encoding is provided.

Each blank is worth one point for a **total of 10 points**. No partial credit.

State Diagram Key	
<b>M<sub>xy</sub></b>	Means move from floor x to floor y. i.e. M <sub>21</sub> means moving from floor 2 to 1
<b>OP<sub>x</sub></b>	Means elevator is on floor x with doors open. i.e. OP <sub>1</sub> means on floor 1 with door open.



```

module elevator
  (clock , reset , B1, B2, B3, HEX0, HEX1);
  input   clock , reset , B1, B2, B3;
  output reg [6:0] HEX0; //floor HEX0[0] = segment 0, etc
  output reg [6:0] HEX1; //door

  parameter O = _____; //O
  parameter C = 7'b1000110; //C
  parameter n1 = 7'b1111001; //1
  parameter n2 = 7'b0100100; //2
  parameter n3 = 7'b0110000; //3

  _____ s , ns , count ;
  reg TR;

  //Tgt3 is count greater than 3 seconds
  //Teq6 is count equal 6 seconds

  _____ Tgt3 , Teq6;

  assign Tgt3 = _____;
  assign Teq6 = (count ==6) ? 1:0;
  
```

```

always @* begin
  case(s)
    OP1: if(B2&T3 | T6) ns = MV12;
          else if(B3&T3) ns = MV13;
          else ns = OP1;
    MV12: ns = OP2;
    MV13: ns = MV23;
    OP2: if(B3&T3) ns = MV23; else if(B1&T3) ns = MV21; else ns = OP2;
    MV21: ns = OP1;
    MV23: ns = OP3;

    OP3: _____;
    _____;
    _____;
  default: ns = OP2;
  endcase
end

always @ (posedge clock) begin
  if (reset == 1'b1)
    s <= OP2;
  else
    _____;
  if (TR || reset)
    _____;
  else if (count < 6)
    count <= count + 1;
  end

always @* begin
  case(s)

    OP1: begin HEX1 = O; HEX0 = n1; TR = 0; end
    MV12:begin HEX1 = C; HEX0 = n1; TR = 1; end
    MV13:begin HEX1 = C; HEX0 = n1; TR = 1; end

    OP2: begin _____; end
    MV21:begin HEX1 = C; HEX0 = n2; TR = 1; end
    MV23:begin HEX1 = C; HEX0 = n2; TR = 1; end
    OP3: begin HEX1 = O; HEX0 = n3; TR = 0; end
    MV32:begin HEX1 = C; HEX0 = n3; TR = 1; end
    MV31:begin HEX1 = C; HEX0 = n3; TR = 1; end

    default: begin HEX1 = O; HEX0 = n2; TR = 0; end
  endcase
end
endmodule

```